

➔ This is the debug to check for flags that sir told:

```
mov ax, 0FFFFh
```

```
; Flag is 0000 0010 0000 0010
```

```
; Carry is 0
```

```
; Parity is 0
```

```
; Sign is 0
```

```
; Zero is 0
```

```
; Auxillary is 0
```

```
; Overflow is 0
```

```
inc ax
```

```
; Flag is 0000 0010 0101 0110
```

```
; Carry is 0
```

```
; Parity is 1
```

```
; Sign is 0
```

```
; Zero is 1
```

```
; Auxillary is 1
```

```
; Overflow is 0
```

```
mov bl, 4
```

```
; Flag is 0000 0010 0101 0110
```

```
; Carry is 0
```

```
; Parity is 1
```

```
; Sign is 0
```

```
; Zero is 1
```

```
; Auxillary is 1
```

```
; Overflow is 0
```

```
sub bl, 5
```

```
; Flag is 0000 0010 1001 0111
```

```
; Carry is 1
```

```
; Parity is 1
```

```
; Sign is 1
```

; Zero is 0

; Auxillary is 1

; Overflow is 0

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0000	SI 0000	CS 19F5	IP 0100	Stack +0 0000	Flags 7202
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0009	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 0 0 0 0

CMD >

0100 B8FFFF	MOV	AX,FFFF
0103 40	INC	AX
0104 B304	MOV	BL,04
0106 80EB05	SUB	BL,05
0109 27	DAA	
010A A20189	MOV	[8901],AL
010D C3	RET	
010E 89D0	MOV	AX,DX

1	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	F0	FE
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00

2

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	00	02	FF	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11	
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0000	SI 0000	CS 19F5	IP 0104	Stack +0 0000	Flags 7254
BX 0000	DI 0000	DS 19F5		+2 20CD	
CX 0009	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 0 1 1 1 0

CMD >

0103 40	INC	AX
0104 B304	MOV	BL,04
0106 80EB05	SUB	BL,05
0109 27	DAA	
010A A20189	MOV	[8901],AL
010D C3	RET	
010E 89D0	MOV	AX,DX
0110 89DA	MOV	DX,BX
0112 EB04	JMP	0118

1	0	1	2	3	4	5	6	7
DS:0000	CD	20	FF	9F	00	EA	F0	FE
DS:0008	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01
DS:0018	01	01	01	00	02	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF
DS:0028	FF	FF	FF	FF	EB	19	C0	11
DS:0030	A2	01	14	00	18	00	F5	19
DS:0038	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00
DS:0048	00	00	00	00	00	00	00	00

2

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
DS:0000	CD	20	FF	9F	00	EA	F0	FE	AD	DE	1B	05	C5	06	00	00
DS:0010	18	01	10	01	18	01	92	01	01	01	00	02	FF	FF	FF	FF
DS:0020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	EB	19	C0	11	
DS:0030	A2	01	14	00	18	00	F5	19	FF	FF	FF	FF	00	00	00	00
DS:0040	05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 le 10 ri

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

AX 0000	SI 0000	CS 19F5	IP 0109	Stack +0 0000	Flags 7295
BX 00FF	DI 0000	DS 19F5		+2 20CD	
CX 0009	BP 0000	ES 19F5	HS 19F5	+4 9FFF	OF DF IF SF ZF AF PF CF
DX 0000	SP FFFE	SS 19F5	FS 19F5	+6 EA00	0 0 1 1 0 1 1 1

CMD >				1							
0106 80EB05	SUB	BL,05		DS:0000	CD 20 FF 9F 00 EA F0 FE						
0109 27	DAA			DS:0008	AD DE 1B 05 C5 06 00 00						
010A A20189	MOV	[89011],AL		DS:0010	18 01 10 01 18 01 92 01						
010D C3	RET			DS:0018	01 01 01 00 02 FF FF FF						
010E 89D0	MOV	AX,DX		DS:0020	FF FF FF FF FF FF FF FF						
0110 89DA	MOV	DX,BX		DS:0028	FF FF FF FF EB 19 C0 11						
0112 EB04	JMP	0118		DS:0030	A2 01 14 00 18 00 F5 19						
0114 31D2	XOR	DX,DX		DS:0038	FF FF FF FF 00 00 00 00						
0116 31C0	XOR	AX,AX		DS:0040	05 00 00 00 00 00 00 00						
				DS:0048	00 00 00 00 00 00 00 00						

2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DS:0000	CD 20 FF 9F 00 EA F0 FE								AD DE 1B 05 C5 06 00 00							
DS:0010	18 01 10 01 18 01 92 01								01 01 01 00 02 FF FF FF							
DS:0020	FF FF FF FF FF FF FF FF								FF FF FF FF EB 19 C0 11							
DS:0030	A2 01 14 00 18 00 F5 19								FF FF FF FF 00 00 00 00							
DS:0040	05 00 00 00 00 00 00 00								00 00 00 00 00 00 00 00							

1 Step	2 ProcStep	3 Retrieve	4 Help ON	5 BRK Menu	6	7 up	8 dn	9 le	10 ri
--------	------------	------------	-----------	------------	---	------	------	------	-------

Q.1.

```

.model small

.stack 100h
.data
msg1 db 10,13, "Enter First number: $"
msg2 db 10,13, "Enter Second number: $"
msg3 db 10,13, "Numbers are equal $"
msg4 db 10,13, "Numbers are not equal $"
.code
main proc
mov ax, @data
mov ds, ax
; Display message to enter the first number
mov dx, offset msg1
mov ah, 09h
int 21h
; Read the first number
mov ah, 01h
; Function to read a character from STDIN
int 21h
; Call DOS interrupt
sub al, 30h
; Convert ASCII to numeric value
mov cl, al
; Store the first number
; Display message to enter the second number
mov dx, offset msg2
mov ah, 09h
int 21h
; Read the second number
mov ah, 01h
; Function to read a character from STDIN
int 21h
; Call DOS interrupt
sub al, 30h
; Convert ASCII to numeric value
mov dl, al
; Store the second number
; Compare the two numbers
cmp dl, cl
je equal
; If equal, jump to label1
; If not equal, print the message
mov dx, offset msg4
mov ah, 09h
int 21h
jmp end_prog
equal:
; If equal, print the message
mov dx, offset msg3
mov ah, 09h
int 21h
end_prog:
mov ah, 76
int 33
main endp

```

end main

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

N
D
Assembling: hellonew.asm
Microsoft (R) Segmented Executable Linker Version 5.31.009 Jul 13 1992
Copyright (C) Microsoft Corp 1984-1992. All rights reserved.

Object Modules [.obj]: hellonew.obj
Run File [hellonew.exe]: "hellonew.exe"
List File [nul.map]: NUL
Libraries [.lib]:
Definitions File [nul.def]:

C:\>hellonew.exe

Enter First number: 1
Enter Second number: 2
Numbers are not equal
C:\>hellonew.exe

Enter First number: 1
Enter Second number: 1
Numbers are equal
C:\>_

```

Q.2.

dosseg

.model small

.stack 100h

.data

array db 10 Dup(?)

.code

main proc

mov ax, @data

mov ds, ax

mov si, offset array

mov cx, 10

; loop

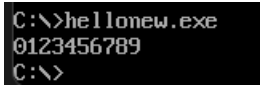
mov al,48

l1:

mov [si],al

mov dx, [si]

```
mov ah,2
int 21h
;mov dx, [si+1]
inc si
inc al
loop l1
mov ah, 4ch
int 21h
main endp
end main
```



A screenshot of a Windows command prompt window. The text displayed is: C:\>hellonew.exe, followed by the output 0123456789 on the next line, and the prompt C:\> on the third line.

Q.3.

```
dosseg
.model small
.stack 100h
.data
array db 10 Dup(?)
.code
main proc
mov ax, @data
mov ds, ax
mov si, offset array
mov cx, 26
; loop
mov al,97
l1:
mov [si],al
mov dx, [si]

mov ah,2
int 21h
```

```
;mov dx, [si+1]
```

```
inc si  
inc al  
loop l1
```

```
mov ah, 4ch
```

```
int 21h
```

```
main endp
```

```
end main
```

```
C:\>hellonew.exe  
abcdefghijklmnopqrstuvwxyz  
C:\>
```

Q.4.

```
dosseg
```

```
.model small
```

```
.stack 100h
```

```
.data
```

```
array db 10 Dup(?)
```

```
.code
```

```
main proc
```

```
mov ax, @data
```

```
mov ds, ax
```

```
mov si, offset array
```

```
mov cx, 26
```

```
; loop
```

```
mov al,65
```

```
l1:  
mov [si],al  
mov dx, [si]
```

```
mov ah,2
```

```
int 21h
```

```
;mov dx, [si+1]
```

```
inc si
```

```

inc al
loop l1


mov ah, 4ch

int 21h

main endp

end main

```



```

C:\>hellonew.exe
ABCDEFGHIJKLMNOPQRSTUVWXYZ
C:\>

```

Q.5.

```

dosseg
.model small
.stack 100h
.data
    var1 db 10,13,'even number..$'
    var2 db 10,13,'odd number..$'
.code
    main proc

        mov ax,@data
        mov ds,ax

        mov ah,1
        int 21h
        mov bl,2

        div bl
        cmp ah,0 ;this si the remainder cehck wether it os 0 or not
        je l1 ;then jump to l1
        mov dx,offset var2 ;else print odd number
        mov ah,9
        int 21h
        jmp exit ;exit

    l1:

        mov dx,offset var1
        mov ah,9
        int 21h
        exit:

        mov ah,4ch
        int 21h

    main endp
end main

```



```

C:\>hellonew.exe
1
odd number..
C:\>hellonew.exe
2
even number..
C:\>_

```

Q.6.

dosseg

.model small

.stack 100h

.data

array db 0,1,2,3,4,5,6,7,8,9

result db 10 Dup(?)

.code

main proc

mov ax, @data

mov ds, ax

mov si, offset array

mov di, offset result

mov cx, 10

l1:

mov al, [si]

mov bl, 2

div bl

cmp ah, 0

;test ah, ah

je evenNumber

mov byte ptr [di], 79

jmp nextElement

evenNumber:

mov byte ptr [di], 101

nextElement:

inc si

inc di

loop l1

mov si, offset result

mov cx, 10

l2:

mov dl, [si]

mov ah, 02h

int 21h

inc si

```

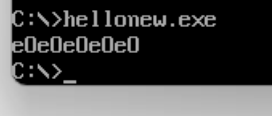
        loop l2 ;

        mov ah, 4ch
        int 21h

main endp

end main

```



```

C:\>hellonew.exe
e0e0e0e0e0
C:\>_

```

Q.7.

dosseg

```
.model small
```

```
.stack 100h
```

```
.data
```

```
array db 2,1,2,1,2
```

```
result db 0
```

```
.code
```

```
main proc
```

```
    mov ax, @data
```

```
    mov ds, ax
```

```
    mov si, offset array
```

```
    mov cx, 5
```

```
    mov ax, 0
```

```
    mov bl, 0
```

```
l1:
```

```
    mov al, [si]
```

```
    add bl, al
```

```
    inc si
```

```
    loop l1
```

```
    mov dl, bl
```

```
    add dl, 48
```

```
    mov result, dl
```

```
    mov ah, 02h
```

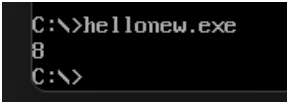
```
    int 21h
```

22i-1552_SYED_ARHAM_AHMED_COAL_LAB#5_BS-CY-B

```
mov ah, 4ch  
int 21h
```

```
main endp
```

```
end main
```



```
C:\>hellow.exe  
B  
C:\>
```

A screenshot of a Windows command prompt window. The window has a black background with white text. The first line shows the command 'C:\>hellow.exe' being entered. The second line shows the output 'B'. The third line shows the prompt 'C:\>' after the command has executed.